WHAT IS CLAIMED IS:

1. A data-processing system for the selective manipulation of data, said data-processing system comprising:

a memory device for storing said data;

an engine having access to said memory device, said engine supporting a plurality of machine executable programs therein;

a controller which selectively outputs one of a plurality of instructions to said engine for driving the execution of said programs enabled by said engine;

a clock device for outputting a synchronizing clock signal comprised of a predetermined number of clock cycles per second, said clock device outputting said synchronizing clock signal to said memory device, said engine and said controller; and

wherein said controller outputs one of said instructions to said engine for execution of one of said programs, while also executing an operation within itself, all within a single clock cycle.

2. A data-processing system, said data-processing system comprising:

a data device for selectively storing data;

an engine having access to said memory device, said engine supporting a plurality of machine executable programs therein;

a controller which selectively outputs one of a plurality of instructions to said engine for driving the execution of said plurality of machine executable programs;

a clock device for outputting a synchronizing clock signal comprised of a predetermined number of clock cycles per second, said clock device outputting said synchronizing clock signal to said data device, said engine and said controller; and

wherein said controller outputs one of said instructions to said engine for execution of one of said plurality of machine executable programs, while also executing an operation within itself, all within one of said clock cycles.

3. The data processing system according to claim 2, wherein:
said data device is an associative memory device having *n*- cells.

4. The data processing system according to claim 3, wherein:
each of said *n*- cells in said associative memory device includes a processing circuit.

5. The data processing system of claim 1, wherein:
said controller includes an execution unit and an integrated data stack having a data stack pointer.

6. The data processing system of claim 5, wherein:
one of said plurality of instructions issued by said controller is a ‘remove’ command whereby a value communicated with said 'remove' command is subtracted from said data stack pointer, thereby popping a number of values equal to said value off the top of said data stack.

7. The data processing system of claim 6, wherein:
one of said plurality of instructions issued by said controller is a ‘restore’ command whereby said value communicated during a most recent execution of said ‘remove’ command is added to said data stack pointer, thereby restoring said data stack pointer to its initial position prior to said execution of said 'remove' command.

8. The data processing system of claim 5, wherein:
said controller further includes a control stack and a control stack pointer.

9. The data processing system of claim 8, wherein:
one of said plurality of instructions issued by said controller is a ‘pushframe’ command whereby a value communicated with said 'pushframe' command is added to said control stack pointer, thereby pushing a number of words equal to a number communicated with said 'pushframe' command onto the top of said control stack.

10. The data processing system of claim 8, wherein:

one of said plurality of instructions issued by said controller is a 'popframe' command whereby a value communicated with said 'popframe' command is subtracted from said control stack pointer, thereby popping a number of values equal to said value off the top of said control stack.

11. The data processing system of claim 8, wherein:

one of said plurality of instructions issued by said controller is a 'xchg' command whereby a top value of said data stack is switched with a top value of said control stack.

12. The data processing system of claim 8, wherein:

one of said plurality of instructions issued by said controller is a 'cs2ds' command whereby a top value of said control stack is pushed onto said data stack.

13. The data processing system of claim 8, wherein:

one of said plurality of instructions issued by said controller is a 'ds2cs' command whereby a top value of said data stack is pushed onto said control stack.

14. The data processing system of claim 8, wherein:

one of said plurality of instructions issued by said controller is a 'pushcs' command whereby a value of said control stack is pushed onto said data stack; and

wherein said value reflects an offset position relative to said control stack pointer, said offset position being equivalent to an amount communicated with said 'pushcs' command.

15. The data processing system of claim 8, wherein:

one of said plurality of instructions issued by said controller is a multi-cycle ‘wait zero’ command whereby a top value of said data stack is compared to zero during one of said clock cycles;

wherein if said top value of said data stack is equal to zero , said top value of said data stack is removed and a program counter is incremented and a subsequent instruction may be executed; and

wherein if said top value of said data stack is not equal to zero, said program counter remains unchanged and said 'wait zero' command is repeated in a subsequent clock cycle.

16. The data processing system of claim 8, wherein:

one of said plurality of instructions issued by said controller is a ‘wait noMarker’ command whereby an internal binary flag indicated by said ‘wait noMarker’ instruction is compared to zero during one of said clock cycles;

wherein if said internal binary flag is equal to zero, a program counter is incremented and a subsequent instruction may be executed; and

wherein if said internal binary flag is not equal to zero, said program counter remains unchanged and said 'wait noMarker' command is repeated in a subsequent clock cycle.

17. The data processing system of claim 8, wherein:

one of said plurality of instructions issued by said controller is a ‘wait noFullOut’ command whereby it is determined, during one of said clock cycles, if an output queue is full;

wherein if said output queue is not full, a program counter is incremented and a subsequent instruction may be executed; and

wherein if said output queue is full, said program counter remains unchanged and said 'wait noFullOut' command is repeated in a subsequent clock cycle.

18. The data processing system of claim 8, wherein:

one of said plurality of instructions issued by said controller is a 'wait noEmptyIn' command whereby it is determined, during one of said clock cycles, if an input queue is empty;

wherein if said output queue is not empty, a program counter is incremented and a subsequent instruction may be executed; and

wherein if said input queue is empty, said program counter remains unchanged and said 'wait noEmptyIn' command is repeated in a subsequent clock cycle.

19. A data-processing system for the selective manipulation of data, said data-processing system comprising:

an associative memory device for storing said data;

an engine having access to said memory device, said engine supporting a plurality of machine executable programs therein;

a controller which selectively outputs one of a plurality of instructions to said engine for driving the execution of said programs enabled by said engine;

a clock device for outputting a synchronizing clock signal comprised of a predetermined number of clock cycles per second, said clock device outputting said synchronizing clock signal to said associative memory device, said engine and said controller; and

wherein said controller outputs one of said instructions to said engine for execution of one of said programs, while also executing an operation within itself, all within a single clock cycle.

20. A method of processing data, said method of processing data comprising the steps of:

storing data in a data device;

providing an engine in communication with said memory device, said engine supporting a plurality of machine executable programs therein;

providing a controller which selectively outputs one of a plurality of instructions to said engine for driving the execution of said plurality of machine executable programs;

outputting a synchronized clock signal having a predetermined number of clock cycles per second to said data device, said engine and said controller; and

outputting one of said plurality of instructions to said engine, and executing one of said plurality of machine executable programs, all within one of said clock cycles.